Bayesian Networks - IV : Probabilistic Relational Models

Philippe LERAY
philippe.leray@univ-nantes.fr

DUKe (Data User Knowledge) Research group
Laboratoire des Sciences du Numérique de Nantes – UMR 6004
Site de l'Ecole Polytechnique de l'université de Nantes

Motivations

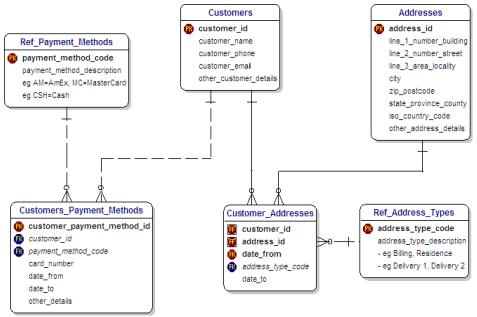
Flat data

- No relational model
- Learning probabilistic dependencies between variables

Motivations

Flat data

- No relational model
- Learning probabilistic dependencies between variables



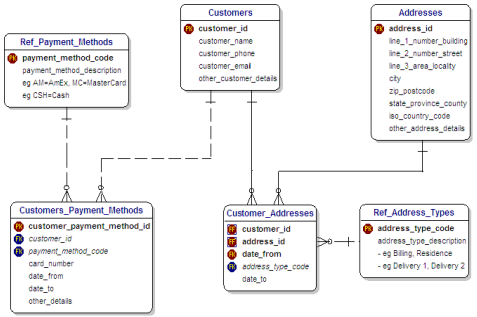
Relational DB

- Relational schema is given
- Learning probablistic dependencies between variables, but more complex !

Motivations

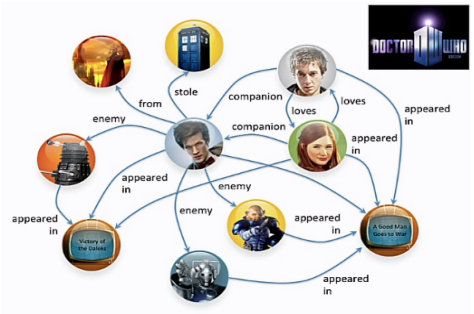
Flat data

- No relational model
- Learning probabilistic dependencies between variables



Relational DB

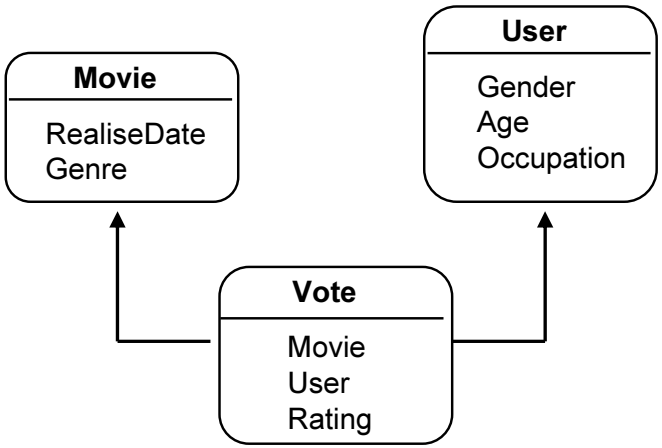
- Relational schema is given
- Learning probablistic dependencies between variables, but more complex !



Graph DB

- Relational schema ?
- Learning probabilistic dependencies between variables ?

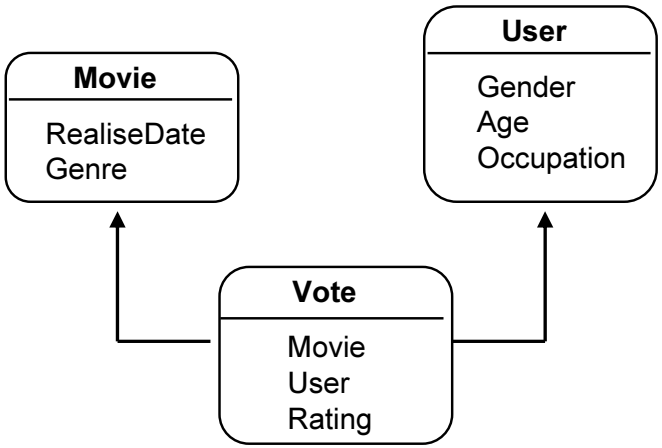
Relational schema $\mathcal{R}$



Definitions

- classes + attributes, $X.A$ denotes an attribute A of a class X
- reference slots = foreign keys (e.g. $Vote.Movie$, $Vote.User$)
- inverse reference slots (e.g. $User.User^{-1}$)
- slot chain = a sequence of (inverse) reference slots

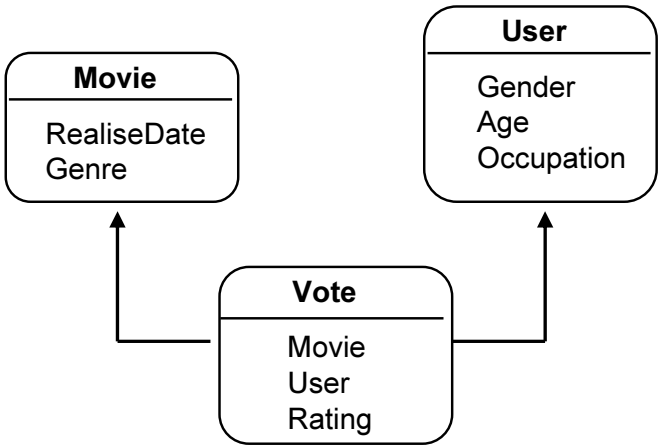
Relational schema $\mathcal{R}$



Definitions

- classes + attributes, $X.A$ denotes an attribute A of a class X
- reference slots = foreign keys (e.g. $Vote.Movie$, $Vote.User$)
- inverse reference slots (e.g. $User.User^{-1}$)
- slot chain = a sequence of (inverse) reference slots
  - ex: $Vote.User.User^{-1}.Movie$: all the movies voted by a particular user

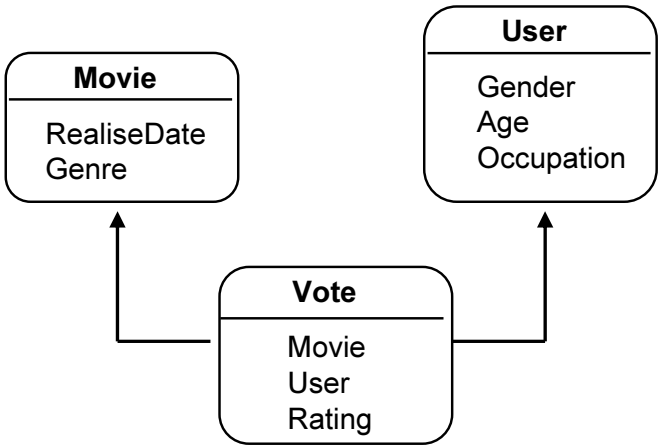
Relational schema $\mathcal{R}$



Definitions

- classes + attributes, $X.A$ denotes an attribute A of a class X
- reference slots = foreign keys (e.g. $Vote.Movie$, $Vote.User$)
- inverse reference slots (e.g. $User.User^{-1}$)
- slot chain = a sequence of (inverse) reference slots
  - ex: $Vote.User.User^{-1}.Movie$: all the movies voted by a particular user

Relational schema $\mathcal{R}$



Definitions

- classes + attributes, $X.A$ denotes an attribute A of a class X
- reference slots = foreign keys (e.g. $Vote.Movie$, $Vote.User$)
- inverse reference slots (e.g. $User.User^{-1}$)
- slot chain = a sequence of (inverse) reference slots
  - ex: $Vote.User.User^{-1}.Movie$: all the movies voted by a particular user

Relational database

Database = an instance of the relational schema

- set of objects for each class
- with a value for each reference slot and each attribute

Movie

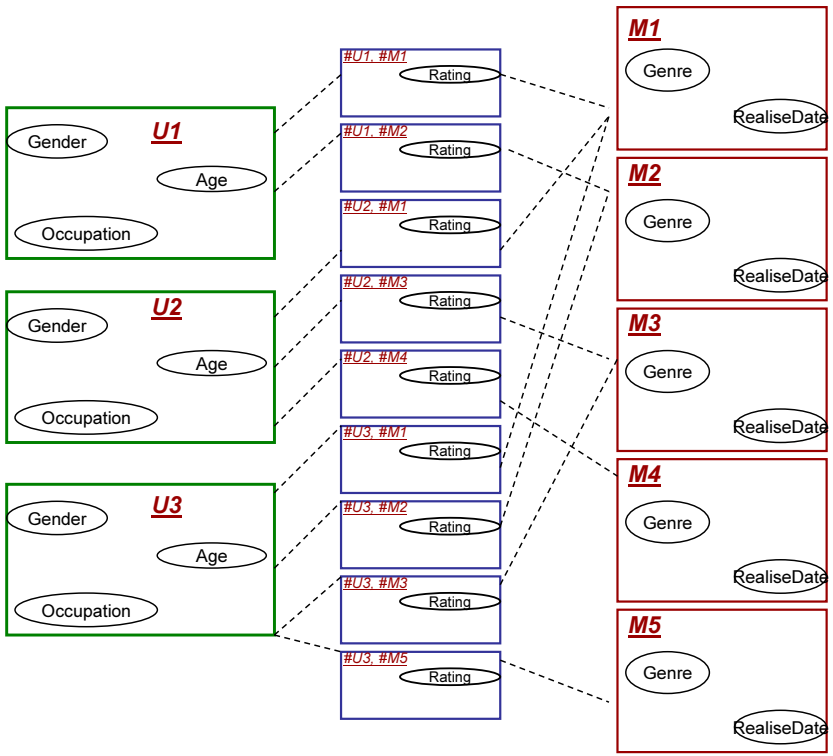
ID	Realise date	Genre
M1	Newest	Action
M2	Vintage	Horror
M3	Current era	Action
...	...	...
MN_1	Vintage	Romance

User

ID	Age	Gender	Occup
U1	Low	F	Student
U2	High	M	Retired
U3	Middle	M	Engineer
...	...	...	...
UN_2	Middle	F	Teacher

Vote

ID	Rating	Movie	User
V1	1	M1	U1
V2	1	M2	U3
V3	4	M2	U1
...	...	...	...
VN_3	2	Mi	Uj

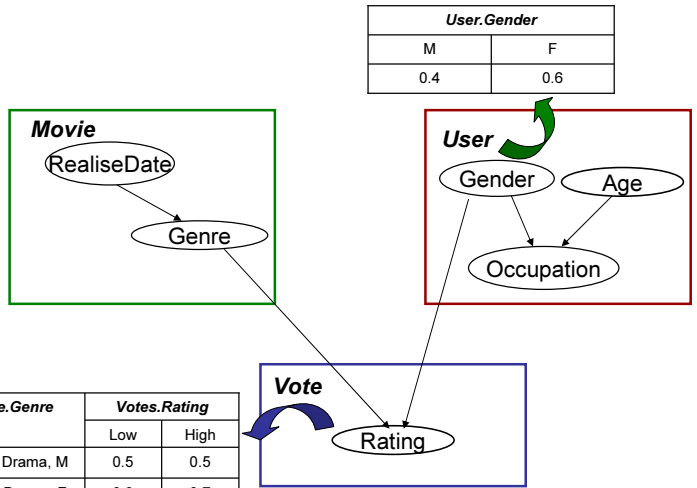


Probabilistic Relational Model (PRM) [Koller & Pfeffer, 1998]

Definition

A PRM Π associated to a relational schema $\mathcal{R}$:

- a qualitative dependency structure $\mathcal{S}$ (with possible long **slot chains** and **aggregation functions**)
- a set of parameters $\theta_{\mathcal{S}}$



Movie.Genre		Votes.Rating	
		Low	High
User.Gender	Drama, M	0.5	0.5
	Drama, F	0.3	0.7
	Horror, M	0.2	0.8
	Horror, F	0.9	0.1
	Comedy, M	0.5	0.5
	Comedy, F	0.6	0.4

Aggregators

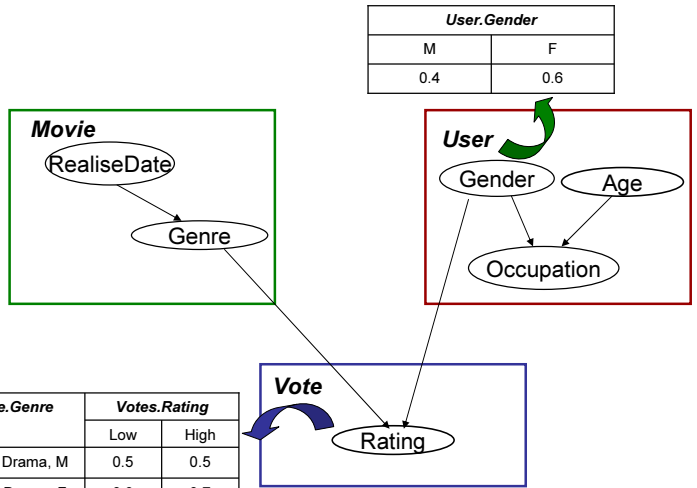
- $Mode(Vote.User.User^{-1}.Movie.genre) \rightarrow Vote.rating$
- movie rating from one user can be dependent with the most frequent genre of movies voted by this user

Probabilistic Relational Model (PRM) [Koller & Pfeffer, 1998]

Definition

A PRM Π associated to a relational schema $\mathcal{R}$:

- a qualitative dependency structure $\mathcal{S}$ (with possible long **slot chains** and **aggregation functions**)
- a set of parameters $\theta_{\mathcal{S}}$



Movie.Genre		Votes.Rating	
		Low	High
User.Gender	Drama, M	0.5	0.5
	Drama, F	0.3	0.7
	Horror, M	0.2	0.8
	Horror, F	0.9	0.1
	Comedy, M	0.5	0.5
	Comedy, F	0.6	0.4

Aggregators

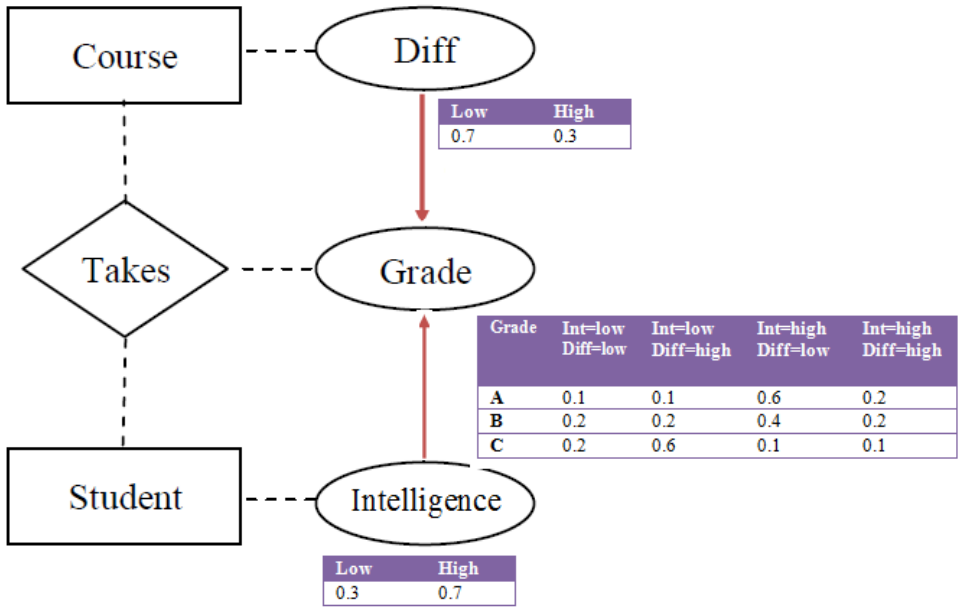
- $Mode(Vote.User.User^{-1}.Movie.genre) \rightarrow Vote.rating$
- movie rating from one user can be dependent with the most frequent genre of movies voted by this user

Directed Acyclic Probabilistic Entity-Relationship model (DAPER)

Another probabilistic relational model [Heckerman & Meek, 2004]

Definition

- Probabilistic model associated to an Entity-Relationship model
- Classes = { Entity classes + Relationship classes }



Low	High
0.7	0.3

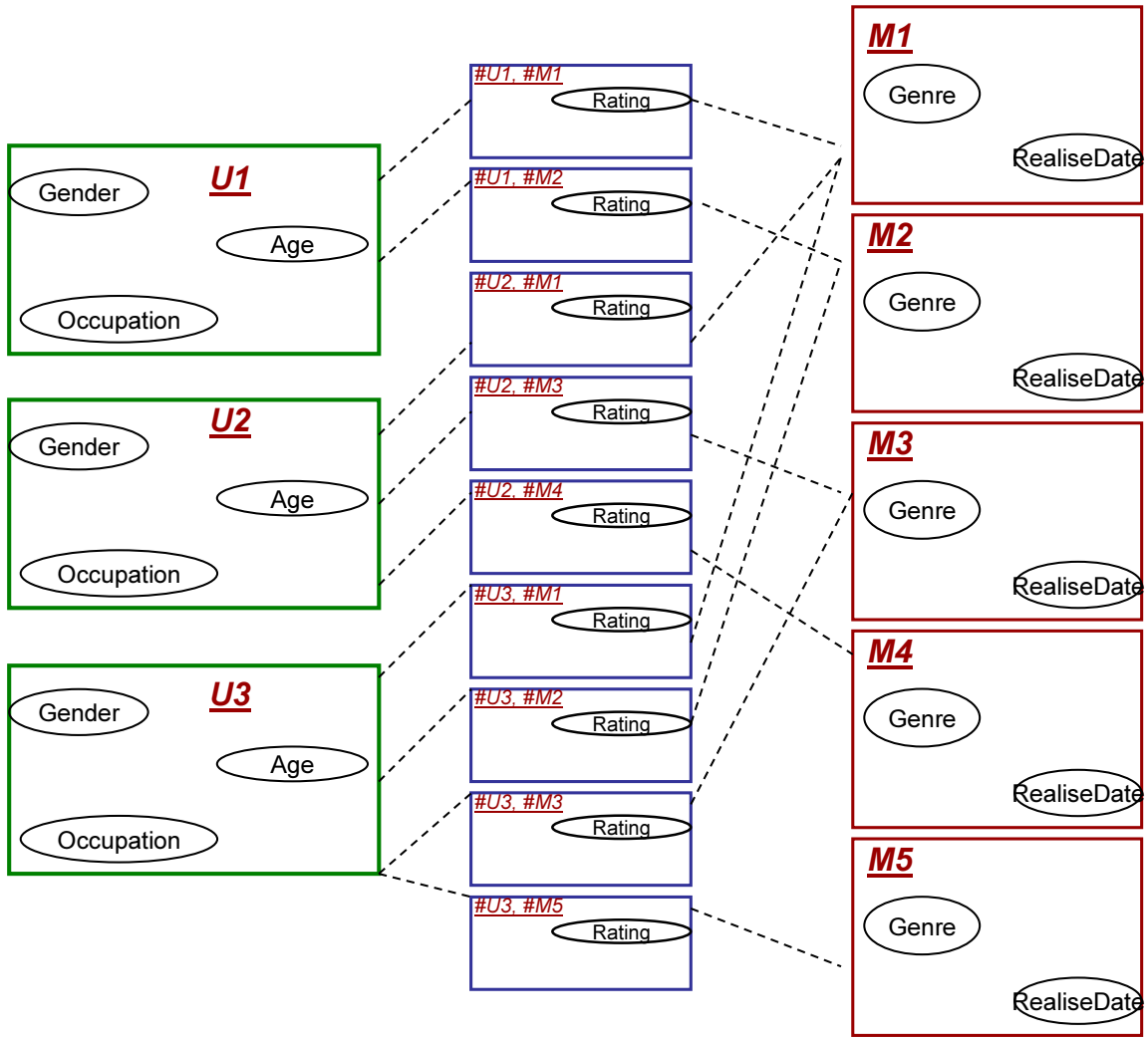
Grade	Int=low Diff=low	Int=low Diff=high	Int=high Diff=low	Int=high Diff=high
A	0.1	0.1	0.6	0.2
B	0.2	0.2	0.4	0.2
C	0.2	0.6	0.1	0.1

Low	High
0.3	0.7

Ground Bayesian Network

GBN

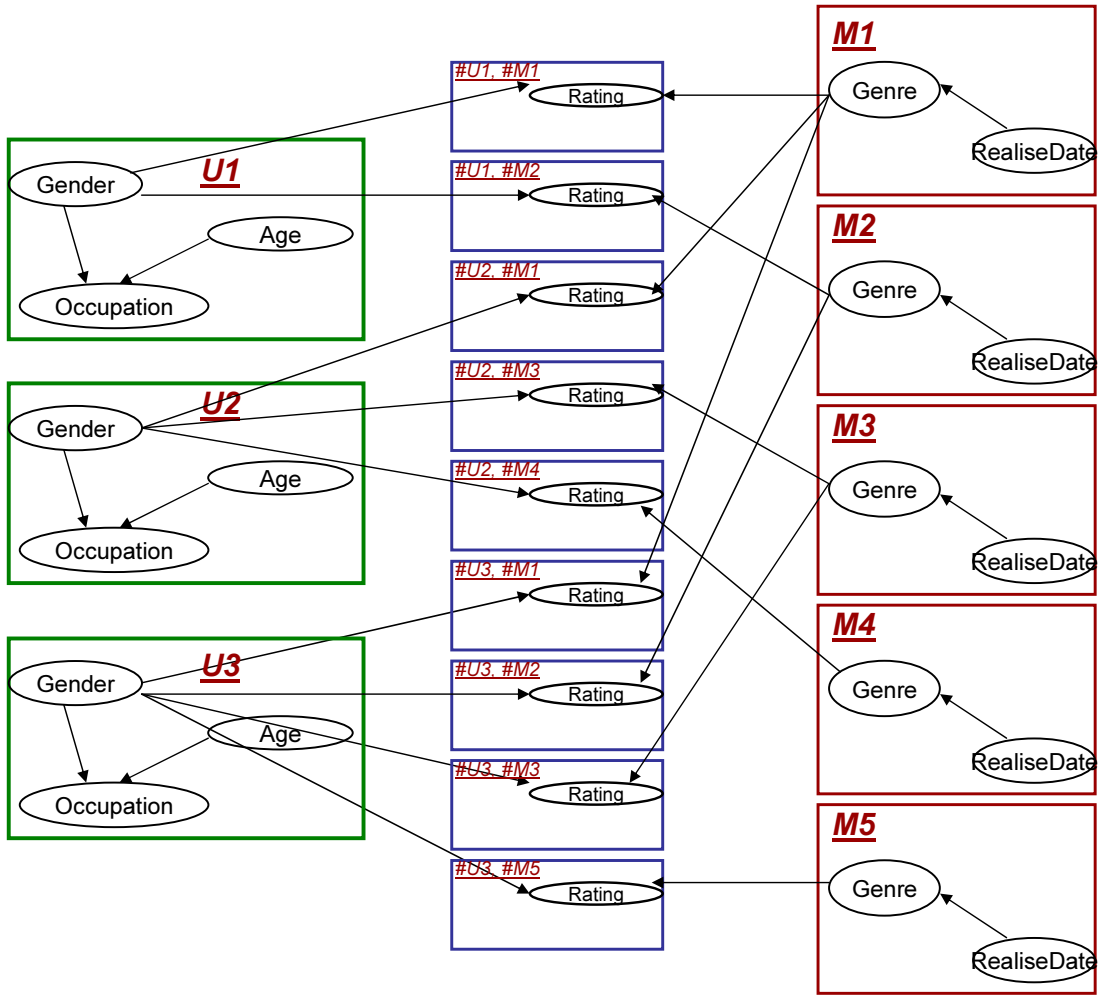
- BN created from one PRM and an instantiated database
- = relational skeleton
- + probabilistic dependencies



Ground Bayesian Network

GBN

- BN created from one PRM and an instantiated database
- = relational skeleton
- + probabilistic dependencies



Inference

Task definition

- given a relational database and a fully defined PRM
- querying the probability of one missing attribute in the database

Solutions

- generating the ground BN, and querying it : complexity issues
- generating a "lazy" ground BN dedicated to the query [Kaelin, 2010]
- using the repetitions or the logical structure induced in the GBN to optimize exact computations [Pfeffer, 1999] [Singla & Domingos, 2008] [Torti, 2012] or sampling [Kaelin, 2010]

Inference

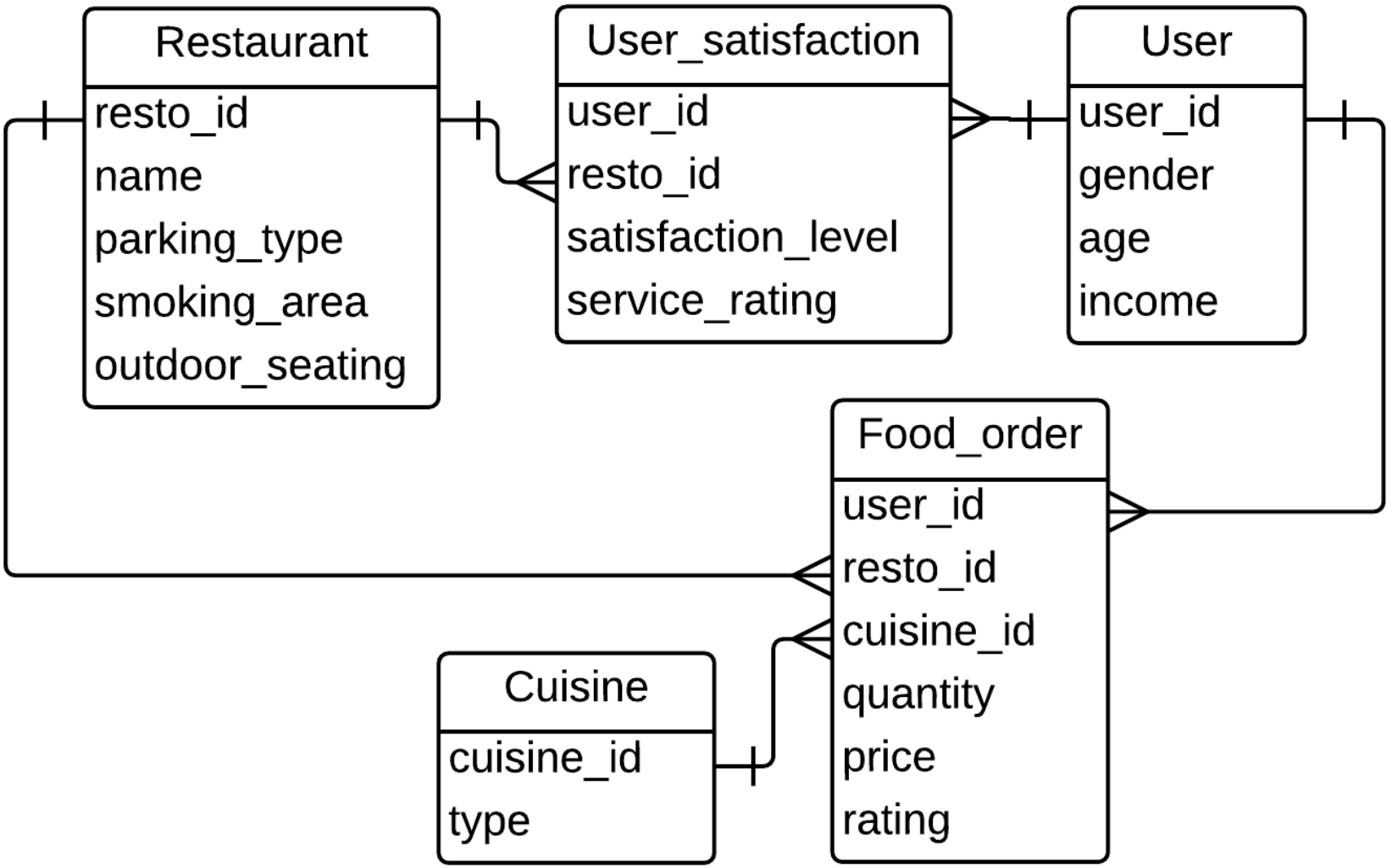
Task definition

- given a relational database and a fully defined PRM
- querying the probability of one missing attribute in the database

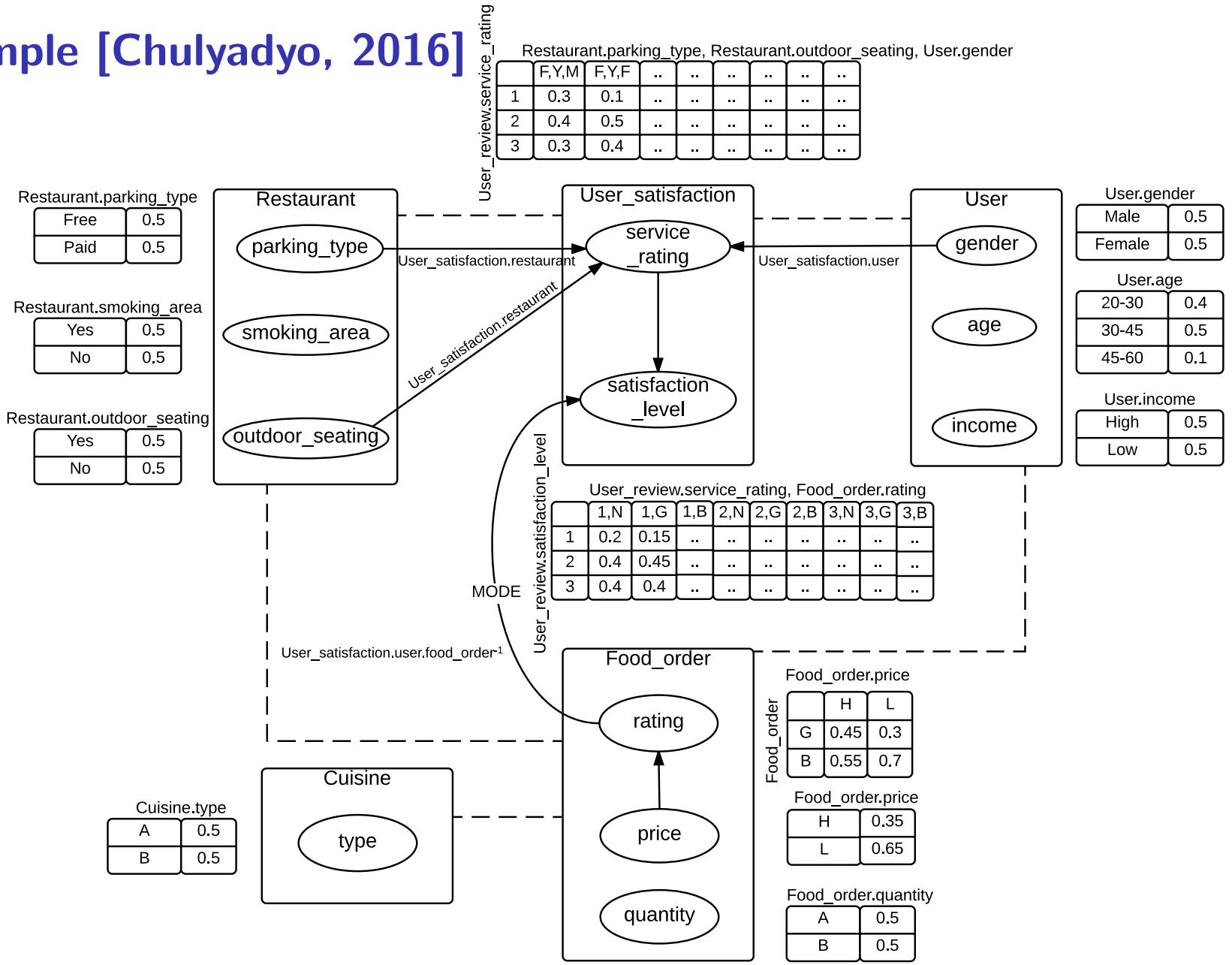
Solutions

- generating the ground BN, and querying it : complexity issues
- generating a "lazy" ground BN dedicated to the query [Kaelin, 2010]
- using the repetitions or the logical structure induced in the GBN to optimize exact computations [Pfeffer, 1999] [Singla & Domingos, 2008] [Torti, 2012] or sampling [Kaelin, 2010]

Example [Chulyadyo, 2016]



Example [Chulyadyo, 2016]



Example [Chulyadyo, 2016]

Restaurant

resto_id	name	parking_type	smoking_area	outdoor_seating
resto_1	ABC	Free	Yes	Yes
resto_2	DEF	Free	No	Yes
resto_3	XYZ	Paid	No	No

Food_order

user_id	resto_id	cuisine_id	quantity	price	rating
user_1	resto_1	cuisine_1	A	H	2
user_1	resto_2	cuisine_2	B	L	3
user_1	resto_1	cuisine_2	A	L	2
user_2	resto_3	cuisine_3	A	L	1

User

user_id	gender	age	income
user_1	Male	20-30	Low
user_2	Female	45-60	High
user_3	Male	30-45	High

Cuisine

cuisine_id	type
cuisine_1	A
cuisine_2	B
cuisine_3	A

User_satisfaction

resto_id	user_id	service_rating	satisfaction_level
resto_1	user_1	1	2
resto_1	user_2	2	2
resto_3	user_1	3	2

- what is the ground BN for this DB ?

PRM parameter learning

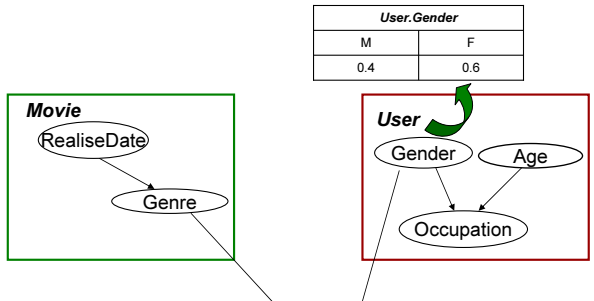
Task definition

- given a relational database and the dependency structure $\mathcal{S}$ (with slot chains and aggregation functions)
- finding the parameters $\theta_{\mathcal{S}}$

Movie		
ID	Realise date	Genre
M1	Newest	Action
M2	Vintage	Horror
M3	Current era	Action
...	...	...
MN_1	Vintage	Romance

User			
ID	Age	Gender	Occup
U1	Low	F	Student
U2	High	M	Retired
U3	Middle	M	Engineer
...	...	...	...
UN_2	Middle	F	Teacher

Vote			
ID	Rating	Movie	User
V1	1	M1	U1
V2	1	M2	U3
V3	4	M2	U1
...	...	...	...
VN_3	2	Mi	Uj



Movie.Genre		Votes.Rating	
		Low	High
User.Gender	Drama, M	0.5	0.5
	Drama, F	0.3	0.7
	Horror, M	0.2	0.8
	Horror, F	0.9	0.1
	Comedy, M	0.5	0.5
	Comedy, F	0.6	0.4

PRM parameter learning

Task definition

- given a relational database and the dependency structure $\mathcal{S}$ (with slot chains and aggregation functions)
- finding the parameters $\theta_{\mathcal{S}}$

What's the difference with BN parameter learning ?

- querying the database in order to count occurrences of events

PRM parameter learning

Task definition

- given a relational database and the dependency structure $\mathcal{S}$ (with slot chains and aggregation functions)
- finding the parameters $\theta_{\mathcal{S}}$

SQL

```
SELECT rating, genre, gender, count(*)
FROM vote, movie, user
JOIN vote ON movie.id = vote.movie-id
JOIN user ON vote.user-id=user.id
GROUP BY rating, genre, gender
```

PRM structure learning

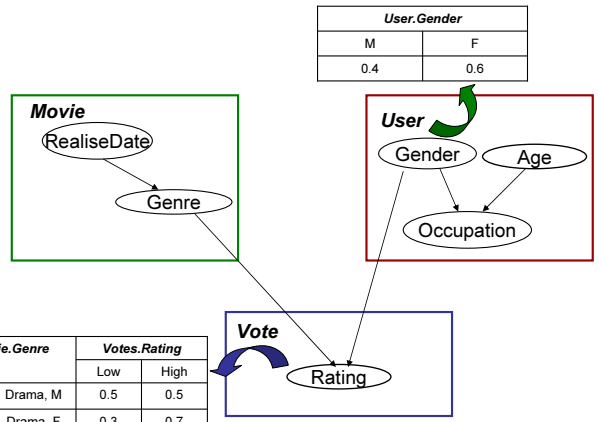
Task definition

- given a relational database
- finding the "meta" probabilistic model (dependency structure $\mathcal{S}$ (with slot chains and aggregation functions) and its parameters $\theta_{\mathcal{S}}$

Movie		
ID	Realise date	Genre
M1	Newest	Action
M2	Vintage	Horror
M3	Current era	Action
...	...	...
MN_1	Vintage	Romance

User			
ID	Age	Gender	Occup
U1	Low	F	Student
U2	High	M	Retired
U3	Middle	M	Engineer
...	...	...	...
UN_2	Middle	F	Teacher

Vote			
ID	Rating	Movie	User
V1	1	M1	U1
V2	1	M2	U3
V3	4	M2	U1
...	...	...	...
VN_3	2	Mi	Uj



Movie.Genre		Votes.Rating	
		Low	High
User.Gender	Drama, M	0.5	0.5
	Drama, F	0.3	0.7
	Horror, M	0.2	0.8
	Horror, F	0.9	0.1
	Comedy, M	0.5	0.5
	Comedy, F	0.6	0.4

PRM structure learning

Task definition

- given a relational database
- finding the "meta" probabilistic model (dependency structure $\mathcal{S}$ (with slot chains and aggregation functions) and its parameters $\theta_{\mathcal{S}}$

What's the difference with BN SL ?

- adding another dimension in the search space (slot chain length x aggregators) + limitation to a given maximal length
- possible existence of multiple dependencies between 2 attributes

PRM structure learning (attribute uncertainty)

Constraint-based methods

- relational PC [Maier et al., 2010] relational CD [Maier et al., 2013], rCD light [Lee and Honavar, 2016]

Score-based methods

- greedy search [Getoor et al., 2007] relational Best first search [Ettouzi et al., 2016]

Hybrid methods

- spanning tree + particle swarm optimization [Li & He, 2014]
- relational MMHC [Ben Ishak et al., 2015]

PRM structure learning (attribute uncertainty)

Constraint-based methods

- relational PC [Maier et al., 2010] relational CD [Maier et al., 2013], rCD light [Lee and Honavar, 2016]

Score-based methods

- greedy search [Getoor et al., 2007] relational Best first search [Ettouzi et al., 2016]

Hybrid methods

- spanning tree + particle swarm optimization [Li & He, 2014]
- relational MMHC [Ben Ishak et al., 2015]

PRM structure learning (attribute uncertainty)

Constraint-based methods

- relational PC [Maier et al., 2010] relational CD [Maier et al., 2013], rCD light [Lee and Honavar, 2016]

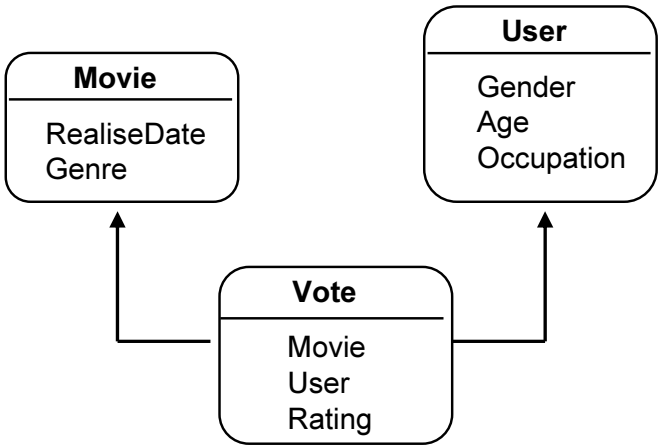
Score-based methods

- greedy search [Getoor et al., 2007] relational Best first search [Ettouzi et al., 2016]

Hybrid methods

- spanning tree + particle swarm optimization [Li & He, 2014]
- relational MMHC [Ben Ishak et al., 2015]

Example : Greedy search [Getoor et al., 2007]



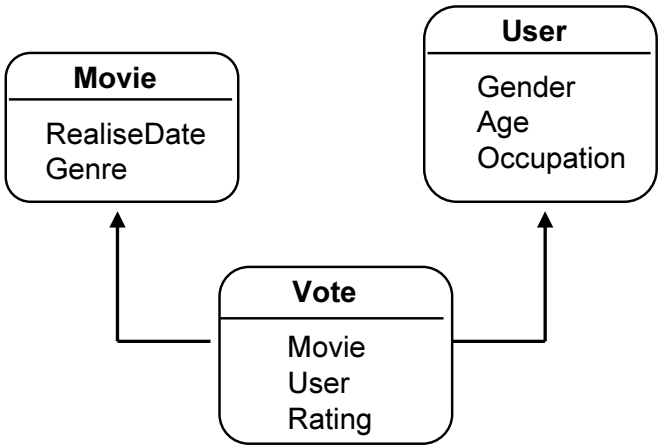
Heuristic

- a BN greedy search embedded in a loop over slot chain length
- a scoring function inspired from BN one

Iteration 0 : possible dependencies in the same class

- M.RealiseDate $\longrightarrow$ M.Genre ? M.Genre $\longrightarrow$ M.RealiseDate ?
- U.Gender $\longrightarrow$ U.Age ? U.Age $\longrightarrow$ U.Gender ? ...

Example : Greedy search [Getoor et al., 2007]



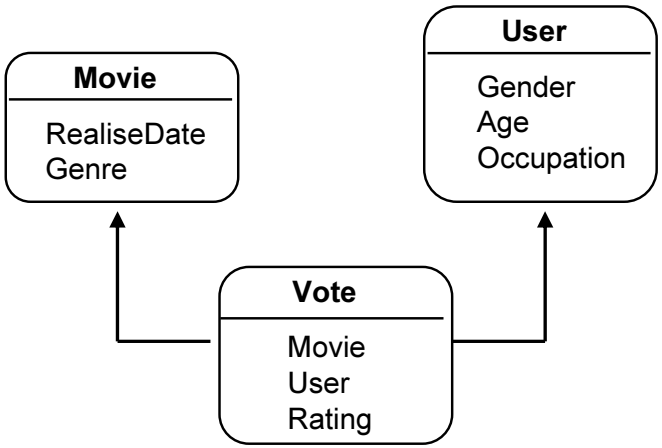
Heuristic

- a BN greedy search embedded in a loop over slot chain length
- a scoring function inspired from BN one

Iteration 1 : possible dependencies in "next" classes

- V.Rating $\longrightarrow$ V.Movie.Genre ?
- M.Genre $\longrightarrow$ mode(M.Movie^{-1}.Rating) ?

Example : Greedy search [Getoor et al., 2007]



Heuristic

- a BN greedy search embedded in a loop over slot chain length
- a scoring function inspired from BN one

And so on ...

- exploring longer and longer slot chains
- but the scoring function penalizes "long" dependencies
- until a maximal slot chain length is reached